|  |  |
| --- | --- |
| **Ex No: 4**  **Date: 06-09-2024** | **CNN Transfer Learning** |

**Objective:**

The primary objective of this project is to build a flower classification model using **transfer learning** with a pre-trained **MobileNetV2** model. The model will be trained to recognize five distinct flower species: **roses, daisy, dandelion, sunflowers,** and **tulips**.

Transfer learning allows us to leverage the knowledge from a model pre-trained on a large dataset (ImageNet) to improve the efficiency and accuracy of the classification task. This approach aims to minimize training time and computational resources while achieving high accuracy. The ultimate goal is to use the trained model for predicting flower species from new images.

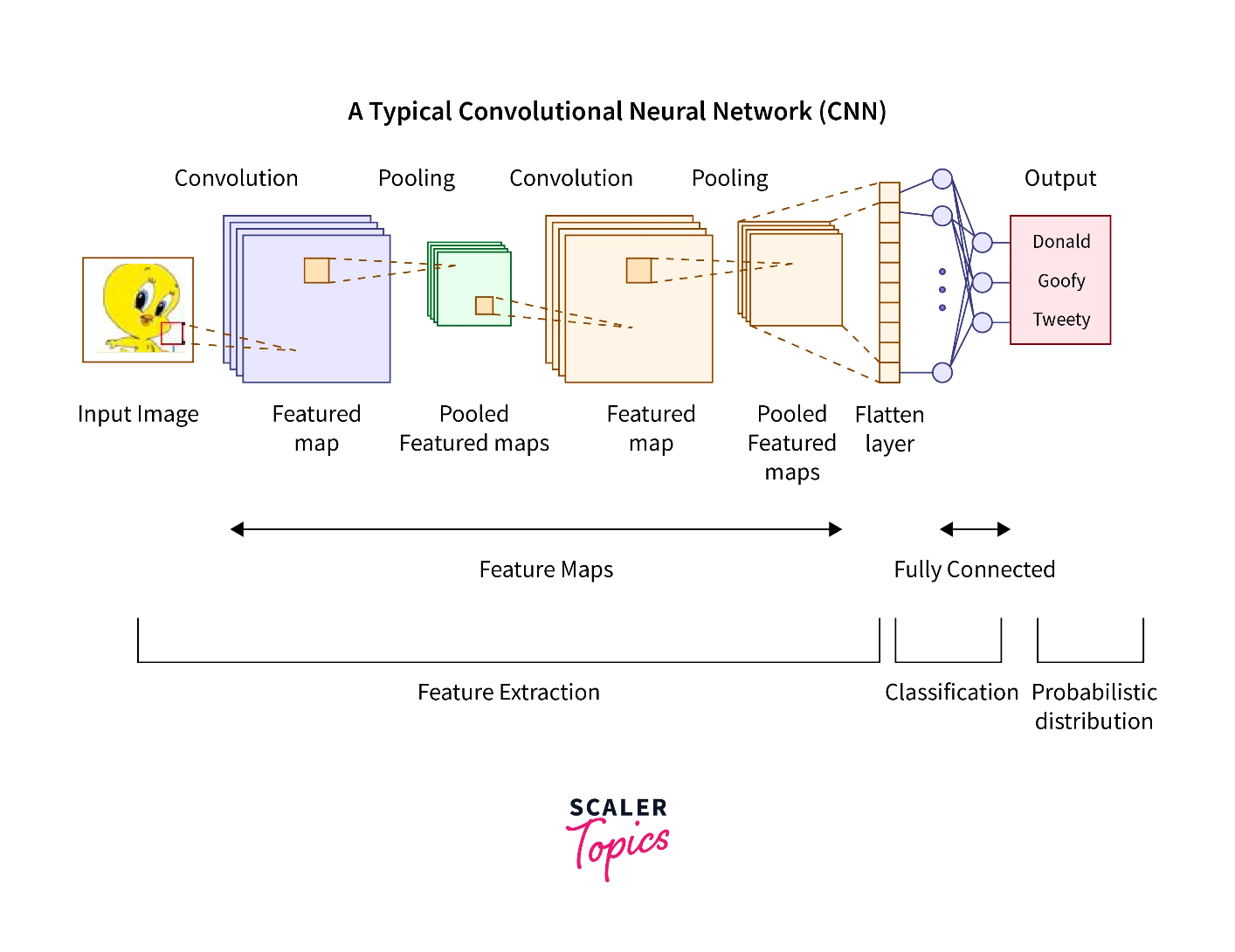
**Descriptions:**

This project uses **TensorFlow** and **TensorFlow Hub** to implement transfer learning for flower classification. The MobileNetV2 model, which was pre-trained on the ImageNet dataset, is used as a feature extractor. By freezing its pre-trained layers, we build a custom classification layer that is fine-tuned to classify images from five categories of flowers: **roses, daisy, dandelion, sunflowers,** and **tulips**.

The flower images are collected from a dataset, which is downloaded and preprocessed into a format suitable for the MobileNetV2 model. The project focuses on resizing the images to a standard input size of **224x224 pixels**, normalizing the data, splitting it into training and test sets, and training a dense layer for classification.

The model is evaluated based on its accuracy in predicting the correct flower type.

**Model:**

****

 **Pre-Trained Model**: MobileNetV2 is used as the feature extractor, which was trained on the large ImageNet dataset. This reduces the need for large amounts of flower-specific data.

 **Transfer Learning**: The lower layers of the pre-trained MobileNetV2 are frozen, and only the final dense layer is trained on the flower dataset, making the training efficient.

 **Input Shape**: All flower images are resized to 224x224 pixels to match the input shape expected by MobileNetV2.

 **Dense Layer**: A custom dense layer with 5 neurons is added, representing the 5 categories of flowers in the dataset.

 **Optimizer**: The model is compiled using the **Adam** optimizer, which is well-suited for handling sparse categorical cross-entropy loss during multi-class classification tasks.

* **Evaluation**: The model is evaluated using accuracy as the primary metric, and its predictions are tested on a separate test dataset.

**Building the parts of algorithm**

* **Dataset Preparation**:
* Download the flower dataset from the TensorFlow website.
* Organize the dataset into five categories: **roses, daisy, dandelion, sunflowers,** and **tulips**.
* Resize all images to **224x224** pixels and normalize pixel values to the range [0, 1].
* **Model Architecture**:
* Use the **MobileNetV2** pre-trained model from TensorFlow Hub as a feature extractor, without the top classification layer.
* Add a fully connected **dense layer** on top for classifying the images into the five flower categories.
* **Training the Model**:
* Split the dataset into training and test sets using **train\_test\_split**.
* Compile the model using the **Adam optimizer** and **Sparse Categorical Crossentropy loss**.
* Train the model for **5 epochs** to classify the flower images.
* **Prediction**:
* After training, use the model to predict the class of new images by preprocessing them into the required format (224x224 pixels, normalized).
* Utilize the **np.argmax** function to determine the class with the highest probability from the model's output.
* **Evaluation**:
* Evaluate the model’s performance on the test set to measure its accuracy and validate its ability to correctly classify unseen flower images.
* Use the trained model to predict and label new flower images.

**GitHub Link:**

[**https://github.com/Chudarathnamani-L/Deep-Learning-Lab-Experiments/tree/master**](https://github.com/Chudarathnamani-L/Deep-Learning-Lab-Experiments/tree/master)

**Ctrl+Click to open this link**